

学校编码: 10384

分类号_____密级_____

学号: 22420051302388

UDC _____

厦门大学

硕 士 学 位 论 文

嵌入式软件单元无关性自动测试
研究与实现

Research and Realization of Automated Unit
Independency Test in Embedded System

黄丽敏

指导教师姓名: 林聪仁 副教授

专 业 名 称: 通信与信息系统

论文提交时间: 2008 年 6 月

论文答辩日期: 2008 年 月

学位授予日期: 2008 年 月

答辩委员会主席_____

评 阅 人_____

2008 年 6 月

嵌入式软件单元无关性自动测试研究与实现

黄丽敏

指导教师: 林聪仁副教授

厦门大学

厦门大学博硕士论文摘要库

厦门大学学位论文原创性声明

兹呈交的学位论文，是本人在导师指导下独立完成的研究成果。
本人在论文写作中参考的其他个人或集体的研究成果，均在文中以明确方式标明。本人依法享有和承担由此论文而产生的权利和责任。

声明人（签名）：

年 月 日

厦门大学学位论文著作权使用声明

本人完全了解厦门大学有关保留、使用学位论文的规定。厦门大学有权保留并向国家主管部门或其指定机构送交论文的纸质版和电子版，有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆被查阅，有权将学位论文的内容编入有关数据库进行检索，有权将学位论文的标题和摘要汇编出版。保密的学位论文在解密后适用本规定。

本学位论文属于

- 1、保密（ ），在 年解密后适用本授权书。
- 2、不保密（√）

（请在以上相应括号内打“√”）

作者签名：

日期： 年 月 日

导师签名：

日期： 年 月 日

摘要

软件测试是保证和提高软件可靠性的重要手段,软件单元测试能最早发现程序错误。但在单元测试中已通过功能测试的子模块(子程序)并不一定是完全正确的。子模块还可能存在一些不易被发现的潜在缺陷,这些缺陷不影响本模块的功能正确性但因资源使用不当而影响其它模块的功能,从而导致系统出错,甚至崩溃。为此本文提出了软件模块无关性的概念,即一个子模块的运行不能影响其它模块及整个系统。软件模块无关性概念包括以下七方面:程序空间无关性、数据空间无关性、堆栈空间无关性、I/O 空间无关性、寄存器无关性、中断无关性、重复调用无关性。

现有的嵌入式软件单元测试工具主要是进行功能性的测试,还没有程序模块无关性测试的功能。针对这种情况,本文对软件模块级无关性自动测试的原理和实现方法进行了探索性的研究,目的是实现模块级功能自动测试和七个方面的无关性自动测试,找到针对不同 CPU 结构的通用测试方法。目前的测试工具与开发工具是分离的两套系统,这不利于尽早发现程序的缺陷,所以我们的最终目标是将该测试工具集成到开发工具中。

本文初步实现了基于 SPCE061A 的软件模块级无关性自动测试工具 **IndependencyTest**, 其功能包括五个方面的无关性测试:数据空间无关性测试、堆栈空间无关性测试、I/O 空间无关性测试、寄存器无关性测试、中断无关性测试。每项无关性测试功能的实现分为四个步骤:被测模块信息的提取、测试方法和测试数据自动构造、测试用例生成及测试结果的输出。对 **IndependencyTest** 工具进行的测试结果表明 **IndependencyTest** 工具能测试出汇编语言程序在无关性方面的潜在编程缺陷。

关键词: 无关性测试; 单元测试; 软件自动测试

Abstract

Software testing is an important means to ensure and improve the software reliability, software unit testing can be the first to detect errors. The sub-module or subprogram which has passed through function test is not always correct. There may be some potential flaws in the sub-module which are difficult to detect. These flaws do not affect the functional correctness of the module, but impact functions of other modules because of improper use of resources, causing system faults or even collapse. This thesis proposes the concept of software unit independency, that a sub-module can't affect the operation of other modules and the entire system. The concept of software module independency includes seven aspects: code space independency, data space independency, stack space independency, I/O independency, registers independency, interruptions independency test, and duplicated calling independency.

The current embedded software unit test tools mainly include function test, but there is not independency test. According to this situation, the thesis has done some explorative research on the principle and realization method of automated software unit independency test. The research objective is to realize the seven aspects of automated independency test and find out the universal testing method of various types of CPU. The current testing tools and development tools are separated each other, which is not avail to detect potential flaws early. So the final goal is to integrate the test tool into development tools.

The thesis primarily realizes the automated unit independency test which based SPCE061A. It includes as following five aspects: data space independency test, stack space independency test, I/O independency test, registers independency test, interruptions independency test. Each function module of independency test includes four steps: the extraction of information, test methods and test data automatically construction, test case generation and the output of the test results. The test results showed that IndependencyTest tool can find out the potential programming flaws of assembly language program.

Key Words: Independency Test; Unit Test; Software Test Automation

目录

第一章 绪论	1
1.1 课题研究的背景	1
1.1.1 软件可靠性.....	1
1.1.2 软件测试的发展现状.....	2
1.1.3 嵌入式软件测试的研究意义.....	3
1.2 课题研究目的和意义	4
1.3 论文的主要工作	5
1.4 论文的结构	6
第二章 软件测试与软件质量	7
2.1 软件质量	7
2.1.1 软件质量.....	7
2.1.2 软件质量保证体系.....	8
2.2 软件测试	10
2.2.1 软件测试过程.....	11
2.2.2 软件测试方法.....	13
2.3 软件自动化测试	14
2.3.1 自动化测试概述.....	14
2.3.2 自动化测试过程.....	15
2.3.3 自动化测试工具.....	17
2.3.4 自动化测试工具的研究现状.....	18
2.4 嵌入式软件测试	20
2.4.1 嵌入式软件测试环境.....	20
2.4.2 嵌入式系统软件测试工具.....	22
2.4.3 嵌入式系统软件测试工具研究现状.....	23
第三章 无关性自动测试工具功能设计	28
3.1 IndependencyTest 工具的工作原理	28
3.2 IndependencyTest 工具的功能设计	29
3.2.1 需求描述.....	29
3.2.2 被测软件模块信息提取模块.....	32
3.2.3 测试方法以及测试数据构造模块.....	32
3.2.4 测试用例产生模块.....	32
3.2.5 测试结果生成模块.....	32
3.3 IndependencyTest 工具的界面设计	33
第四章 基于 SPCE061A 的 IndependencyTest 工具的实现	35
4.1 开发平台简介	35
4.1.1 SPCE061A 单片机简介	35
4.1.2 SPCE061A 单片机硬件结构	37
4.1.3 μ^{TM} nSP™汇编语言程序设计.....	44
4.2 IndependencyTest 工具的被测信息提取模块实现	47
4.2.1 测试子程序的语法规则.....	47
4.2.2 被测模块信息提取.....	49
4.3 IndependencyTest 工具的功能实现	50

4.3.1 数据空间无关性测试功能的实现.....	50
4.3.2 堆栈空间无关性测试功能的实现.....	52
4.3.3 I/O 空间无关性测试功能的实现.....	53
4.3.4 寄存器无关性测试功能的实现.....	56
4.3.5 中断无关性测试功能的实现.....	58
4.4 测试结果输出模块实现	60
第五章 IndependencyTest 工具的测试	62
5.1 测试环境	62
5.2 IndependencyTest 工具的测试	62
5.2.1 测试内容.....	62
5.2.2 测试方案.....	62
5.2.3 测试结果.....	65
第六章 总结和展望	69
6.1 总结	69
6.2 展望	69
[参考文献].....	70
致谢	73
读研期间发表论文	74

Contents

Chapter 1 Introduction.....	1
1.1 Background of Research Subject	1
1.1.1 Software Reliability	1
1.1.2 Current Situation of Software Testing.....	2
1.1.3 Purpose of Research Embedded Software Testing.....	3
1.2 Purpose of Research Subject	4
1.3 Major Work of the Thesis.....	5
1.4 Arrangement of the Thesis.....	6
Chapter 2 Software Testing and Software Quality.....	7
2.1 Software Quality.....	7
2.1.1 Software Quality.....	7
2.1.2 Software Quality Assurance System.....	8
2.2 Software Testing.....	11
2.2.1 Software Testing Process.....	11
2.2.2 Software Testing Methods.....	13
2.3 Automated Software Testing.....	14
2.3.1 Summary of Automated Software Testing.....	14
2.3.2 Automated Software Testing Process.....	15
2.3.3 Automated Software Testing Tool.....	17
2.3.4 Current Situation of Automated Software Testing Tool.....	18
2.4 Embedded Software Testing.....	20
2.4.1 Environment of Embedded Software Testing.....	20
2.4.2 Testing Tool of Embedded Software.....	22
2.4.3 Current Situation of Embedded Software Testing Tool.....	23
Chapter 3 Design Functions of IndependencyTest Tool.....	28
3.1 Principle of IndependencyTest Tool.....	28
3.2 Design of IndependencyTest Tool.....	29
3.2.1 Requirements Description.....	29
3.2.2 Module of Information Extraction.....	32
3.2.3 Module of Testing Methods and Test Data Automatic Construction	32

3.2.4 Module of Test Cases Generation.....	32
3.2.5 Module of Test Results Generation.....	32
3.3 Interface Design of IndependencyTest Tool.....	33
Chapter 4 Realization of Independency Test Tool.....	35
4.1 Brief Introduction of Development Platform.....	35
4.1.1 Brief Introduction of SPCE061A.....	35
4.1.2 Hardware Structure of SPCE061A.....	37
4.1.3 Assembly Language Programming of SPCE061A.....	44
4.2 Realization of Information Extraction Module.....	47
4.2.1 Syntax Rule of Target Subprogram.....	47
4.2.2 Information Extraction of Target Subprogram.....	49
4.3 Realization of IndependencyTest Tool.....	50
4.3.1 Realization of Data Space Independency Test.....	50
4.3.2 Realization of Stack Space Independency Test.....	52
4.3.3 Realization of I/O Independency Test.....	53
4.3.4 Realization of Registers Independency Test.....	56
4.3.5 Realization of Interruptions Independency Test.....	58
4.4 Realization of Test Results Generation Module.....	60
Chapter 5 Testing of IndependencyTest Tool.....	62
5.1 Test Configuration.....	62
5.2 Testing of IndependencyTest Tool.....	62
5.2.1 Test Items.....	62
5.2.2 Test Method.....	62
5.2.3 Test Results.....	65
Chapter 6 Summary and Improvement.....	69
6.1 Summary.....	69
6.2 Improvement.....	69
References.....	70
Acknowledgement.....	73
Postscript.....	74

第一章 绪论

1.1 课题研究的背景

1.1.1 软件可靠性

由于计算机技术和软件技术的飞速发展,软件规模日益增大、日益复杂。20 世纪 60 年代中期,国外就进入所谓的“软件危机时期”。IBM 公司开发 OS/360^[1] 系统投入 5000 人/年,耗资数亿美元,结果还是延期交付,在交付使用后的系统中仍发现大量(2000 个以上)的错误。1988 年^[2],超大规模集成电路出现,但是,由软件引发的、震惊科技界的故障持续不断,在某些类型的设备中软件故障甚至远远超过了硬件,成为系统的主要故障源。美国 Therac 25 型放射治疗仪 2# 治疗模式(x 射线模式)发生的 54#故障,多次产生超计量辐射,造成了两人死亡和多人受伤的重大医疗事故。Therac 25 是 Therac 6 的改进型,使用软件安全互锁装置,旧式的 Therac 6 使用机械安全互锁装置,却从来没有发生过类似故障。90 年代后半期,“千年虫”^[3]问题震惊世界,各国投入了大量的人力和物力,耗资数千亿美元,才基本得到控制。“千年虫”实际上就是一种特殊的软件故障。

可见,软件可靠性已成为软件技术发展的突出问题,这就要求人们在软件工程领域以及整个计算机工程领域中进一步考虑和重视软件可靠性问题。

国外从 60 年代后期进入软件危机时期就开始加强软件可靠性的研究工作,经过 20 年左右的研究推出了各种可靠性模型和预测方法,于 1990 年前后形成较为系统的软件可靠性工程体系。同时,从 80 年代中期开始^[4],西方各主要工业强国均确立了专门的研究计划和课题,如英国的 AIVEY(软件可靠性和度量标准)计划、欧洲的 ESPRIT(欧洲信息技术研究与发展战略)计划^[5]、SPMMS(软件生产和维护管理保障)课题^[6]、Eureka(尤里卡)计划等。目前国际上,软件可靠性发展比较有影响的学术会议有:软件可靠性工程国际会议(ISSRE)、软件维护国际会议(ISSM)和软件保证、确认国际会议(ISACC)等。

国内软件可靠性研究始于 80 年代初,黄锡滋等在软件可靠性建模、软件可靠性分配及软件可靠性管理等方面作了有益的探索^[7];姚一平等在软件可靠性建

模、软件可靠性评估工具和混合硬件-软件系统可靠性等方面作了努力^[9]；刘纯生在软件容错方面作了重要工作^[10]；蔡开元在软件可靠性建模(提出模糊软件可靠性模型)、软件可靠性模型应用选择、软件可靠性测试、软件可靠性度量体系作了有益的探讨^[11]。由于起步较晚，国内在软件可靠性量化理论、度量标准（指标体系）、建模技术、设计方法、测试技术等方面与国外差距较大。目前，国内也越来越重视软件的可靠性问题，国家为了推动这方面的研究，提供了很多项目资助，如 2007 年国家自然科学基金委员会启动“可信软件基础研究”重大研究计划等^[12]。

1.1.2 软件测试的发展现状

软件测试是提高和保证软件可靠性的重要手段，提高软件可靠性是软件测试的主要目的之一。

1975 年，Goodenough 和 Gerhart 定义了测试可靠性模型，奠定了软件测试的理论基础。此后，Howden, Geller 和 Gourlay 等人相继探讨了路径测试的可靠性、软件测试在程序正确性证明上的应用以及软件测试的数学模型的问题，促使测试理论深入发展。总的来说，软件测试技术的发展经历了三个阶段^[12]：

1. 初级阶段（60 年代初-70 年代初）。这个阶段的主要工作是个别方法的探索 and 不同方法的收集，这个时期的代表著作是 Hetzel, B. 的《The Complete Guide to Software Testing》。
2. 发展阶段（70 年代初-70 年代末）。这个阶段的主要工作是逐步建立了软件测试的理论基础，代表著作是 Myers, G. J. 的《The Art of Software Testing》。
3. 成熟阶段（70 年代末至今）。软件测试进入了工程化、标准化和规范化阶段，建立了各种测试工具和环境，丰富和完善了软件测试的理论和方法，并在软件开发中得到广泛应用，使软件测试成为改进和提高软件质量有效和不可缺少的方法。代表著作是 Beize, B. 的《Software Testing Techniques》与《Software System Testing and Quality Assurance》。

目前，较常用的软件测试工具有：Ration Teststudio 测试套件、Winrunner、LoadRunner 和 Quick Test Professional（简称 QTP）等等。

就目前软件工程发展的状况而言，软件测试仍然是较为薄弱的方面，其

方法和工具亟待提高。结合软件的开发实践,软件测试的发展趋势有以下几方面:

1. 测试与开发相融合,成为一种缺陷预防的有效方法。软件测试与开发过程的其他阶段不再是串行工作方式,而是与整个开发过程并行进行。针对开发全过程所开展的软件测试和过程度量,注重根据对测试数据的统计分析结果,来判断软件产品的未来质量趋势,并提前予以控制和预防,属于一种“防患于未然”的质量管理方式。

2. 软件需要进行易测试性设计。软件易测试性设计即是在软件的设计和编码中考虑测试问题,软件易测试性设计的目的是在不增加或者少增加软件复杂性的基础上,将易于测试的原则融合到设计和编码之中。软件易测试性设计体现软件测试的如下观点:软件产品的可靠性建立在分析、设计、编码、测试整个流程之上,而不是仅仅依靠软件测试来保障。软件易测试性设计也体现了软件测试与软件开发的设计和编码阶段相融合的发展趋势。

3. 自动化测试必将更充分、更频繁的被利用。软件测试中需要大量的机械的、重复的、非智力的工作,所以自动化测试的引入和推广是不可逆转的趋势。

4. 软件测试必须与不断出现的新技术相适应。软件开发技术日新月异,从最初的面向过程,到面向对象到现在的组件式开发;网络的应用把单纯的单机应用扩展到分布式计算,开发方式也从传统的瀑布模式到现在的 XP、RUP。各种新的开发技术层出不穷,相应的对每一个新的开发技术必然要求有新的测试技术与之相适应。

1.1.3 嵌入式软件测试的研究意义

随着嵌入式硬件的飞速发展,嵌入式系统软件的规模日益增大,其结构也越来越复杂。另外嵌入式系统往往应用在通讯、工控、航天、军事等对产品质量要求非常严格的领域,用于一些关键性任务的控制管理。所以,由它产生的错误往往会导致无可挽回的、致命的损失。这就使得对软件系统的测试对嵌入式系统的可靠性显得尤为重要。

由于嵌入式系统的自身特点,如专用性,实时性,内存不丰富,I/O 通道少,开发工具昂贵,并且与硬件紧密相关 CPU 种类繁多等。嵌入式软件的开发和测试也就与一般商用软件的开发和测试策略有了很大的不同,可以说嵌入式软件是最

难测试的一种软件。

嵌入式系统的开发过程是一个软硬件互相协调，互相反馈和互相测试的过程。嵌入式软件对硬件的依赖性要求，软件测试时必须最大限度地模拟被测软件的实际运行环境，以保证测试的可靠性。底层程序和应用程序界限的不清晰增加了测试时的难度。系统测试时，错误的定位较为困难。

由于嵌入式软件设计是以一定的目标硬件平台为基础的、面向固定的任务进行的，因此，一旦被加载到目标系统上，功能必须完全确定。这个特点决定了嵌入式应用软件的继承性较差，延长的系统的测试时间，增加了测试费用。

嵌入式系统中，应用软件自身算法的复杂度和操作系统任务调度，决定了系统资源的分配和消耗，因此，对系统实时性进行测试时，要借助一定的测试工具对应用程序算法复杂度和操作系统任务调度进行分析测试。

综上，可见嵌入式软件与传统的面向对象的和面向过程的软件相比有其自身的特点。针对这些特点对嵌入式软件的测试进行研究是有必要的，有意义的。

1.2 课题研究目的和意义

软件单元质量是软件质量保证的基础。单元测试的目标是特定模块的函数或代码块，这种测试涉及到程序内部设计和代码实现细节，所以通常测试是由开发工程师自己来做。如果有一个单元测试的框架，它的使用就像编译器一样简单，每次修改代码后，用一个命令来进行单元自动测试，就可以尽早发现程序中的潜在缺陷。

目前，同一般软件测试工具相比，嵌入式软件测试工具相对比较少，而嵌入式软件单元自动化测试工具更少。绝大多数测试工具的主要功能是功能及性能的测试。软件功能或软件性能都是比较直观的软件问题，测试中比较容易定位程序出错的地方。但软件子模块还可能存在一些不易被发现的潜在缺陷，这些缺陷不影响本模块的功能正确性但因资源使用不当而影响其它模块的功能，从而导致系统出错，甚至崩溃。为此本文提出了软件模块无关性的概念，即一个子模块的运行不能影响其它模块及整个系统。软件模块无关性概念包括以下七个方面：程序空间无关性、数据空间无关性、堆栈空间无关性、I/O 空间无关性、寄存器无关性、中断无关性、重复调用无关性。因软件潜在缺陷而导致的错误是很难发现和

定位的，所以更应该从软件系统的最底层，即从子程序模块入手消除软件无关性隐患。

从嵌入式微控制器（单片机）、嵌入式微处理器到嵌入式 DSP，其软件开发环境的功能越来越完善，如汇编、编译、代码优化、软硬件仿真、运行时间统计等等，这些功能都集成在一个开发环境中，但这些开发环境都还没有软件自动测试功能。众所周知，软件测试应该同软件开发周期的各个阶段紧密结合，但目前的测试工具与开发工具是分离的两套系统，单独使用嵌入式软件测试工具，费用较为昂贵，而且不方便。若能将无关性自动测试及功能自动测试工具集成在软件开发环境中，不仅能在软件开发阶段就自动查出软件的无关性错误，而且有利于程序员养成规范的编程习惯。这是本文提出的软件模块级无关性自动测试工具研究课题的最终目标。

本课题研究的意义在于提出了软件单元测试中的无关性测试概念，并强调应该将无关性自动测试及功能自动测试工具集成到开发工具中，大大拓展了嵌入式软件测试的思路。

1.3 论文的主要工作

本文的主要工作包括：

- 1、收集软件可靠性、软件测试和嵌入式软件测试的相关资料，总结国内外软件测试的发展现状，分析在软件调试阶段进行软件无关性测试的意义。
- 2、分析软件测试、软件自动化测试、嵌入式软件测试的基本方法、基本过程，及现有的各种软件测试工具的特点及其应用领域。
- 3、研究软件模块级无关性自动测试工具的基本工作原理，设计软件模块级无关性自动测试工具的模型，实现系统的被测模块信息提取、测试方法和测试数据自动构造、测试用例生成以及测试结果的输出等关键模块的功能。
- 4、研究设计基于凌阳 16 位单片机 SPCE061A 的汇编语言程序的模块级无关性自动测试工具 `IndependencyTest` 工具。其主要功能是实现汇编语言子程序在数据空间无关性、堆栈空间无关性、I/O 空间无关性、

寄存器无关性、中断无关性，五个方面的无关性测试。反复测试 `IndependencyTest` 工具，系统可以正常执行指定的任务，运行稳定。运用 `IndependencyTest` 工具汇编子程序，测试结果均能说明子程序在无关性方面存在的编程缺陷。

1.4 论文的结构

本论文的结构安排如下：

第一章，概况介绍软件可靠性和软件测试的发展及研究；分析了嵌入式软件测试的研究意义；阐述研究课题的背景和研究意义；最后给出论文各部分的内容安排。

第二章，介绍软件质量的保证体系，软件测试和软件自动化测试的基本过程和方法，分析了软件自动化测试的发展以及国内外软件自动化测试工具的研究现状；分析了嵌入式软件测试的特点及国内外嵌入式软件测试研究现状。

第三章，介绍软件模块级无关性自动测试工具 `IndependencyTest` 的功能设计，包括 `IndependencyTest` 工具的工作原理和利用 `Visual C++6.0` 编写的工具界面，`IndependencyTest` 工具的实现的四大功能模块：被测程序信息的提取、测试方法和测试数据自动构造、测试用例生成以及测试结果的输出。

第四章，详细阐述了基于凌阳 16 位单片机 `SPCE061A` 的 `IndependencyTest` 工具的具体实现。简单介绍了软件的开发平台，详细阐述了 `IndependencyTest` 工具的五种测试功能的具体实现：数据空间无关性测试功能、堆栈空间无关性测试功能、I/O 空间无关性测试功能、寄存器无关性测试功能、中断无关性测试功能。

第五章，利用 `IndependencyTest` 工具对若干 $\mu^n\text{SP}^{\text{TM}}$ 汇编语言子程序进行五项无关性测试，给出测试结果并分析。

第六章，总结本文的工作，并对未来的研究工作进行展望。

Degree papers are in the "[Xiamen University Electronic Theses and Dissertations Database](#)". Full texts are available in the following ways:

1. If your library is a CALIS member libraries, please log on <http://etd.calis.edu.cn/> and submit requests online, or consult the interlibrary loan department in your library.
2. For users of non-CALIS member libraries, please mail to etd@xmu.edu.cn for delivery details.

厦门大学博硕士论文摘要库